Dokumentácia projektu č.2 z IPK

ARP Scanner

Autor : Peter Grofčík

Login : xgrofc00

23.4.2017

Obsah

1 Zadanie

Úlohou projektu bolo vytvoriť ARP scanner lokálnej siete pomocou RAW BSD socketov. Lokálna sieť pre scanner vychádza zo zadaného rozhrania, pričom výstupom je vygenerovaný XML súbor s názvom z argumentu príkazovej riadky.

./ipk-scanner -i interface -f file

2 Implementácia

2.1 ipk-scanner.cpp

Po spustení programu dôjde k vyriešeniu správnosti argumentov a k následnému zápisu základnej hlavičky do zadaného súboru. Následne dôjde k zavolaniu funkcie frame (v request.cpp). Vygenerovanú štruktúru arp requestu (z funkcie frame) využívam pri posielaní požiadaviek na jednotlivé ip adresy v sieti. Ip adresy na ktoré je požiadavka posielaná sa odvíjajú od ip adresy rozhrania(zistenej vo funkcii frame). Pomocou ip adresy rozhrania a jeho masky zistím ip adresu siete a požiadavku teda posielam s ip adresou zariadenia z rozsahu (adresa siete < target < broadcast). V momente odoslania requestu dôjde k vytvoreniu potomka hlavného procesu ktorý následne očakáva odpoveď z danej ip adresy. Očakávanie odpovede je limitované 10 ms. (Z pohľadu vlastného testovania je daná hodnota dostačujúca na prijatie odpovede v prípade že na danej ip adrese sa nachádza zariadenie. Jediný problém nastal pri android zariadení pripojenom cez wifi adaptér a to len v prípade opätovného spustenia programu do určitého času po jeho ukončení resp. keď požiadavka bola odoslaná viac krát za sebou, tak odpovedal iba prvý krát a nejaký čas po odpovedi android nereagoval na opätovnú požiadavku). Po prijatí odpovede automaticky zapisujem do súboru (XML) informácie o ip adrese a mac adrese zariadenia.

2.2 request.cpp

2.2.1 Arp

Jedná sa o štruktúru arp požiadavku, ktorú som vytvoril na základe špecifikácii arp protokolu z wiki stránky a pomocou referenčnej literatúry *An Ethernet Address Resolution Protocol*

ARP wiki

Internet Protocol (IPv4) over Ethernet ARP packet		
octet offset	**0**	**1**
0	Hardware type (HTYPE)	
2	Protocol type (PTYPE)	
4	Hardware address length (HLEN)	Protocol address length (PLEN)
6	Operation (OPER)	
8	Sender hardware address (SHA) (first 2 bytes)	
10	(next 2 bytes)	
12	(last 2 bytes)	
14	Sender protocol address (SPA) (first 2 bytes)	
16	(last 2 bytes)	
18	Target hardware address (THA) (first 2 bytes)	
20	(next 2 bytes)	
22	(last 2 bytes)	
24	Target protocol address (TPA) (first 2 bytes)	
26	(last 2 bytes)	

2.2.2 Frame

Funkcia vracia štruktúru arp požiadavku vytvorenú z časti staticky a z časti pomocou informácii ktoré získa z rozhrania zadaného ako tretí argument programu. Pričom však nenastaví destination adresu (k jej nastaveniu dochádza až v hlavnej časti programu). Funkcia tiež nastavuje dve premenné typu uint8_t na hodnotu masky a broadcastu zisteného z rozhrania.

2.2.3 Errs

Jednoduchá funkcia na chybné ukončenie programu volaná z rôznych častí programu s rôznou hodnotou (int)parametra odpovedajúcou návratovej hodnote programu. Funkcia pred ukončením programu vypíše chybovú hlášku s popisom chyby ktorá nastala pri behu programu.

3 Abort signál

Správnosť formátu XML súboru som vyriešil odchytom abort signálu (ctrl +c). Po jeho odchyte je globálna premenná typu volatile sig_atomic_t nastavená na hodnotu true. V takomto prípade dôjde v hlavnej časti programu k zápisu chýbajúceho formátovania XML súboru hlavným procesom a k ukončeniu programu.

4 Demonštrácia

K odpovedi zo zariadenia nemusí nutne dôjsť hneď po požiadavke. Odpoveď očakáva vedľajší proces programu po určitú dobu od odoslania požiadavky hlavným procesom.

```
isa2015@isa2015:~/Documents/IPK$ sudo tcpdump -i eth0 -v "arp"
tcpdump: listening on eth0, link-type EN10MB (Ethernet), capture size 65535 bytes
15:47:12.298007 ARP, Ethernet (len 6), IPv4 (len 4), Request who-has 192.168.1.129 tell 192.168.1.132, length 46
15:47:12.298431 ARP, Ethernet (len 6), IPv4 (len 4), Reply 192.168.1.129 is-at f8:1a:67:63:d1:46 (oui Unknown), length 46
15:47:12.298838 ARP, Ethernet (len 6), IPv4 (len 4), Request who-has 192.168.1.130 tell 192.168.1.132, length 46
15:47:12.316104 ARP, Ethernet (len 6), IPv4 (len 4), Request who-has 192.168.1.131 tell 192.168.1.132, length 46
15:47:12.330768 ARP, Ethernet (len 6), IPv4 (len 4), Request who-has 192.168.1.132 tell 192.168.1.132, length 46
15:47:12.342763 ARP, Ethernet (len 6), IPv4 (len 4), Request who-has 192.168.1.133 tell 192.168.1.132, length 46
15:47:12.354715 ARP, Ethernet (len 6), IPv4 (len 4), Request who-has 192.168.1.134 tell 192.168.1.132, length 46
15:47:12.354921 ARP, Ethernet (len 6), IPv4 (len 4), Reply 192.168.1.134 is-at 88:ae:1d:eb:d0:c0 (oui Unknown), length 46
15:47:12.355245 ARP, Ethernet (len 6), IPv4 (len 4), Request who-has 192.168.1.135 tell 192.168.1.132, length 46
15:47:12.366866 ARP, Ethernet (len 6), IPv4 (len 4), Request who-has 192.168.1.136 tell 192.168.1.132, length 46
15:47:12.379783 ARP, Ethernet (len 6), IPv4 (len 4), Request who-has 192.168.1.137 tell 192.168.1.132, length 46
15:47:12.390871 ARP, Ethernet (len 6), IPv4 (len 4), Request who-has 192.168.1.138 tell 192.168.1.132, length 46
15:47:12.402688 ARP, Ethernet (len 6), IPv4 (len 4), Request who-has 192.168.1.139 tell 192.168.1.132, length 46
15:47:12.415982 ARP, Ethernet (len 6), IPv4 (len 4), Request who-has 192.168.1.140 tell 192.168.1.132, length 46
15:47:12.426713 ARP, Ethernet (len 6), IPv4 (len 4), Request who-has 192.168.1.141 tell 192.168.1.132, length 46
15:47:12.429811 ARP, Ethernet (len 6), IPv4 (len 4), Reply 192.168.1.131 is-at a0:8d:16:77:bb:4e (oui Unknown), length 46
15:47:12.430161 ARP, Ethernet (len 6), IPv4 (len 4), Request who-has 192.168.1.142 tell 192.168.1.132, length 46
15:47:12.439848 ARP, Ethernet (len 6), IPv4 (len 4), Request who-has 192.168.1.143 tell 192.168.1.132, length 46
15:47:12.451579 ARP, Ethernet (len 6), IPv4 (len 4), Request who-has 192.168.1.144 tell 192.168.1.132, length 46
15:47:12.462627 ARP, Ethernet (len 6), IPv4 (len 4), Request who-has 192.168.1.145 tell 192.168.1.132, length 46
15:47:12.474662 ARP, Ethernet (len 6), IPv4 (len 4), Request who-has 192.168.1.146 tell 192.168.1.132, length 46
15:47:12.487322 ARP, Ethernet (len 6), IPv4 (len 4), Request who-has 192.168.1.147 tell 192.168.1.132, length 46
```

5 Bibliografia

1. https://en.wikipedia.org/wiki/Address_Resolution_Protocol
2. https://en.wikipedia.org/wiki/Berkeley_sockets
3. https://tools.ietf.org/html/rfc826